社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－ARC－169 (1)
2006／7／31

PCクラスタにおける全体電力プロファイルを用いた電力性能最適化

堀田義彦† 佐藤三久† 木村英明†
朴泰祐† 高橋大介†

近年，従来低消費電力向けプロセッサに実装されていた，消費電力を削減するためにプロセッサの動作周波数・電圧を動的に変更するDVFS(Dynamic Voltage and Frequency Scaling)が高性能プロセッサにも実装されている．性能低下を抑え，消費電力を削減するためには通信やメモリアクセスの際に適切な周波数スケジューリングを行う必要がある．我々はこれまでに消費電力の測定を行い，プロファイル情報と消費電力プロファイルを使用したDVFSの最適化の提案を行ってきた．しかし，大規模なクラスタシステムにおいて，電力性能を最適化するDVFSスケジューリングを行うのに必要な各ノードの消費電力プロファイルを取得することは困難である．そこで本論文では，PCクラスタのシステム全体消費電力プロファイルを用いたDVFSスケジューリングを行う電力性能最適化手法を提案する．各ノードの詳細な消費電力特性を取得するため，システム全体の消費電力プロファイルとプログラムをいくつかの領域に分割し，実行することで得られる実行プロファイルを元に各ノードの消費電力プロファイルの推測手法を提案し，評価を行った．得られた各ノードの消費電力プロファイルを使用し，DVFSのオーバーヘッドを考慮した周波数選択アルゴリズムを適用することで，標準の周波数で動作するときと比べ，エネルギーを最大9%，電力遅延積であるEDPを最大約8%削減できることがわかった．

Power Performance Optimization using Total Power Profile on a PC cluster

YOSHIHIKO HOTTA,† MITSUHISA SATO,† HIDEAKI KIMURA,†
TAISUKE BOKU† and DAISUKE TAKAHASHI †

Currently, several high performance processors used in a PC cluster have a DVS (Dynamic Voltage Scaling) architecture that can dynamically scale processor voltage and frequency. Adaptive scheduling of the voltage and frequency enables us to reduce power dissipation without a performance slowdown during communication and memory access. It is difficult to measure the power consumption characteristics of all nodes for selecting adaoptive DVFS scheduling on a conventional large scale HPC cluster system.

In this paper, we propose a method of power-performance optimization by using total power consumption characteristics of the system. We estimate power consumption characteristics of each node from total power consumption and profile-information of each node. By the estimated power consumption characteristics of each node and execution profile, we obtain DVFS scheduling using optimization algorithm to select a *gear* with taking into account for DVFS transition overhead. We examined the effectiveness of our method on Power-Scalable cluster. As the results of benchmark tests, we achieved almost 9% reduction of energy compared to that using the standard clock frequency.

1. はじめに

近年，PCクラスタなどの高性能計算機システムにおいてプロセッサの消費電力の上昇による消費電力の増加が問題となっている．プロセッサの消費電力は100Wを越え，冷却のために大きな装置が必要となりシステムのコンパクト化の大きな障害となっている．また，高い消費電力のためにプロセッサの高密度実装も困難になっている．この消費電力の爆発的な上昇は地球シミュレータやASCIシリーズのような巨大なシステムの設計において非常に大きな問題として認識された．この問題のひとつの解決方法は，低消費電力プロセッサを用いることである．BlueGene/L[1]では低消費電力なコンポーネントを使用することでシステム全体の消費電力を削減し，高密度実装による高性能化を実現している．また，Fengらによる Green

† 筑波大学大学院システム情報工学研究科
Graduate School of Information and Sciences Engeneering,University of Tsukuba

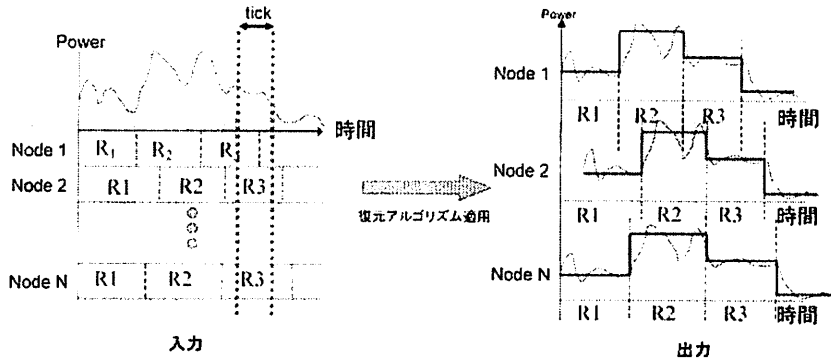

図 1　消費電力推測の概念図

Destiny[5] は低消費電力プロセッサを高密度に実装することで高性能を実現し，低消費電力化の必要性を提言した最初のクラスタである．我々は Transmeta の Efficeon プロセッサを使用し数千以上のプロセッサで構成される mega scale computing を実現するためのプラットホームとして MegaProto を開発している[4]．

一方で，消費電力を削減する仕組みとしてプロセッサの電圧・周波数を動的に変化させることで消費電力を下げる DVFS(Dynamic Voltage and Frequency Scaling) があり，現在様々なプロセッサに実装されている．

これまでに，我々は DVFS を HPC クラスタシステムにおいて，最適に利用するために，消費電力測定環境である PowerWatch を構築し，各ノードの消費電力と実行プロファイル情報を用いて消費電力を最適化する技法について提案してきた[6]．最適化において，DVFS を高性能計算に使用するにあたり，問題となる周波数変更のオーバーヘッドを考慮したスケジューリングアルゴリズムを提案し，OS や LongRun[3] などのプロセッサによる DVFS の自動制御と比較して，わずかな性能低下で大幅な消費電力の削減をすることができる．しかし，この手法では各ノードの消費電力の測定が必要になり，今後更なる大規模化が予想される高性能計算機システムへ適用することは困難である．

このような問題を解決するために，我々は全体消費電力プロファイルを元にした消費電力削減手法を提案する．各ノードの消費電力の測定が物理的に困難であってもシステム全体の消費電力を一箇所で測定することは十分に可能である．提案する手法は，このシステム全体の消費電力から各ノードの消費電力を推測し，求められた消費電力プロファイルを元に，最適な DVFS スケジューリングを求めることによって消費電力の削減を行うものである．

提案する手法を評価するために，DVFS が使用可能な 16 ノードのクラスタとシステム全体の消費電力を測定するための交流用消費電力測定コンポーネントを PowerWatch に追加した．提案手法を適用した結果，全体で約 9%のエネルギーを削減することができた．

本論文の構成は，以下の様になる．2 章では，我々が提案する全体消費電力を元にした消費電力最適化について述べる．3 章では，測定に使用した環境について述べる．4 章では評価実験の結果について述べる．5 章で考察を述べ，最後にまとめと今後を示す．

2. 全体電力プロファイルを用いた最適化

2.1 電力性能の指標

本論文では PDP (Power Delay Product)，EDP (Energy Delay Product) の両方の指標で評価を行う．PDP は単位時間の消費電力と実行時間の積分によって求められ，アプリケーションを実行するのに必要なエネルギーの量を表し，その単位を Ws とする．我々の対象は高性能かつ低消費電力なクラスタであり，このことをふまえて評価するには PDP だけでは十分ではなく，性能を加味した指標での評価が必要である．そこで，電力遅延積である EDP を使用することにした．EDP は PDP より，実行時間で重みを付けた指標であり，HPC クラスタにおけるエネルギー効率を評価するために使用されている．本論文において，EDP を以下のように定義する．

$$EDP = T_{exectime} \times Energy$$

この指標において，EDP の値が低いことは優れた電力性能でプログラムを実行することを意味する．

2.2 全体消費電力による最適化

電力性能を最適化する周波数スケジューリングを行うには，各ノード消費電力プロファイルが必要になる．しかし，大規模並列システムにおいて各ノードの消費電力の測定を行うことは物理的に困難である．そこで，本論文ではシステム全体の消費電力を元に各ノードの消費電力を推測する手法を提案する．

2.2.1 消費電力プロファイル推測手法

図 1 に消費電力推測を行う際の入力と出力の関係を示す．入力として，各ノードの実行プロファイル情報とシステム全体の消費電力プロファイルをあたえる．

出力は，各ノードの消費電力プロファイルならびに各領域の平均消費電力とする．アプリケーションの実行が全てのノードで同一であるならば，全体の消費電力をノード数で割ることによって，各ノードの消費電力プロファイルを取得することができる．しかし，並列アプリケーションにおいて通信などのタイミングや計算量の相違，ノードによって異なる仕事を実行することが頻繁にある．

このように，ノードによって実行しているフェーズが異なる場合，単純に割ることによって各ノードの消費電力プロファイルを取得することはできない．

まず，各ノードの消費電力を推測する前提として，

(1) 各領域は同一の消費電力で実行される
(2) 推測された消費電力の合計はシステム全体の消費電力プロファイルと同一とする
(3) 消費電力はなくある一定間隔 *tick* 毎に測定を行う

を設定する．

アプリケーションを N 個の領域に分けるとして，n 番目の領域を R_n と表記する．また，領域 R_i を実行するときの平均消費電力を P_i とする．P_i に関する N 個の方程式を解くことで，各領域の平均消費電力を求める．このとき，k 番目の方程式は

$$\sum_{i=1}^{N} \alpha_{ik} P_i = \beta_k \qquad (1)$$

とする．また *tick* の間隔を Δt とする．j 番目の tick について考える．このときに，領域 R_i を実行している時間を t_i とすると，

$$\sum_{i=1}^{N} t_{ij} \times P_i = p_j \times \Delta t \qquad (2)$$

となる．p_j は j 番目の tick の消費電力である．

ここで，方程式を解くだけならば，N 個の tick を取り上げて，N 個の方程式とすることもできるが，前提条件 (2) により，平均消費電力を求めるときに使用するエネルギーはアプリケーション全体と同一でなければならないため単純に N 個の tick から方程式を作ることはできない．そのため，全ての tick 取り上げ，組み合わせることで N 個の方程式を作成する．しかし，一つの方程式に対して取り上げる tick の選択によっては，方程式が非独立となり解を求めることができない．

そこで式 (1) の k 番目の方程式は，R_k を実行している時間が最大となる tick の式 (2) の合計として構成する．これによって，P_i に関する N 個の方程式を作ることができる．ちなみに，このとき $\sum \beta_k$ は，全体の消費電力になる．

この方程式を解くことで各領域の平均消費電力が求まる．求められた各領域の平均消費電力を用いて，各ノードの実行プロファイルと対応させることで各ノードの消費電力プロファイルの推測を行う．

2.2.2 周波数選択アルゴリズム

ここでは，我々が提案してきた DVFS のオーバーヘッドを考慮した周波数選択アルゴリズムに付いて説明する．各領域ごとに最適な周波数を決定するために，評価関数を用意した (R_i (i = 1 ···))

$$E(P) = \sum_{i=1}^{n} (E(R_i, f_i) + E_{trans}(R_i, f_i))$$

評価指標はどのようなものでも良いが，EDP として説明する．この式において，$E(R_i, f_i)$ は領域 R_i を動作周波数 f_i で実行するときの EDP の総和である．f_i はプロセッサが変更することができる *gear* の周波数である．$E_{trans}(R_i, f_i)$ は *gear* を動作周波数 f_i で実行するときの周波数変更に必要なオーバーヘッドの際の EDP の総和を表している．目的は $E(P)$ を最小にする f_i の組み合わせを求めることである．プログラム上の領域の先頭で，周波数を制御するため同一の領域は同じ周波数で実行されるものとする．このアルゴリズムは，周波数が決まっていない領域の中から $E(R_i, f_{max})$ が最大となる領域から実行を開始し，$E(P)$ を最小にする f_i を求める．その際に隣接領域と異なる周波数を評価する場合は，DVFS のオーバーヘッドを加味して評価を行う．アルゴリズムの詳細に関しては我々の先行研究[6] に詳しく記してある．

2.2.3 全体消費電力による最適化の手順

この手法は，以下の手順により実行を行う

(1) **プロファイル取得のための試行**
プロファイル情報を付加したアプリケーションを使用可能な全ての周波数で実行することで各領域の実行時間のプロファイルを取得する．また，同時にシステム全体の消費電力プロファイルを取得する．

(2) **各ノードの消費電力プロファイルの推測**
実行プロファイルと全体電力プロファイルを元に各ノードの消費電力プロファイルの推測を行う．

(3) **周波数選択アルゴリズムを適用**
周波数選択アルゴリズムを実行し，各領域における最適な周波数を選択する．

(4) **実際に周波数スケジューリングを適用**
求められた最適な周波数を適用する．プログラム内の各領域の直前に DVFS のシステムコールを挿入し，求められた周波数で各領域を実行する．

3. 電力測定環境 PowerWatch

本論文における評価実験は，消費電力と実行時間のプロファイルを取得するためのシステム PowerWatch[6] を使用する．今回，システム全体の合計消費電力を求

表 1 クラスタの仕様

システム名	Turion cluster	Cecily cluster
CPU	Turion MT-32	Opteron148
Clock	1.6GHz	2.2GHz
Memory	1GB(DDR)	1GB(DDR)
OS (kernel)	2.6.7	2.6.15
MPI	LAM 7.1.1	LAM 7.1.1
Num of nodes	8	16
Network	Gigabit Ethernet	Gigabit Ethernet

表 2 Opteron の周波数と電圧

gear	周波数	電圧	TDP
1	1000MHz	1.10V	36.1W
2	1200MHz	1.15V	–
3	1400MHz	1.20V	–
4	1600MHz	1.25V	–
5	1800MHz	1.30V	70.4
6	2000MHz	1.35V	83.0
7	2200MHz	1.40V	85.3W

めるために PowerWatch に交流電源を測定するためのコンポーネントを追加開発した．これは，複数の交流電源をひとつにまとめ，複数ノードの合計の消費電力の測定を行うことができるコンポーネントである．これにより，システム全体の合計消費電力の測定や各ノードの実行プロファイルを取得することができる．アプリケーションの実行プロファイルを取得するためにイベントの実行時間を取得するライブラリである tlog を使用した．tlog とは *time log* の略称であり，イベントごとの log を取るライブラリと可視化を行うツールである tlogview から構成されている．プロファイルを取得するために，プロファイル取得コードをプログラム内の適当な場所 (例えば MPI_Send など) を挟みこむことで，tlog 初期化からの開始時点の経過時間と終了時点の経過時間を取得することができる．

4. 評価実験

4.1 消費電力推測の精度

提案する各ノードの消費電力の推測が正しく機能するかどうか，あらかじめ各ノードの消費電力を測定することができるクラスタを用いて，消費電力推測手法の予備評価を行った．表 1 に使用した Turion クラスタの仕様を示す．Turion cluster の各ノードの消費電力プロファイルを合計し，システム全体の消費電力を作成した．そのプロファイルをと各ノードの実行プロファイルを用いることで各ノードの消費電力プロファイルを求め，全ノードの合計を行い，実測との比較を行った．図 6 に NPB FT のメインループ実行時のある区間の推測と実測の比較を示す．このときの動作周波数は 1.8GHz である．消費電力推測手法では，各領域を実行する平均消費電力が求まるため，実測と比較して，中間の値を取る傾向がある．領域の中で消費電力が高いところと低いところが存在することも影響している．消費電力の変化の傾向は十分に推測することができており，このプロファイル情報を用いて周波数選択アルゴリズムを適用することで，電力性能最適化を十分に行うことができる．

4.2 クラスタ環境

評価実験を行うために，DVFS を使用することができる Opteron を使用した 16 ノードのクラスタを構築した．表 1 に構築したクラスタの仕様を示す．表 2 に Opteron148 プロセッサで使用可能な周波数と電圧の組合せを示す．最適周波数選択アルゴリズムを使用するのに必要な周波数変更のオーバーヘッドは 50 μ 秒とした．

表 3 各ベンチマークの領域

prog	領域 1	領域 2	領域 3
IS	rank()	MPI_allreduce	MPI_Alltoallv
FT	fft()	evolve()	MPI_Alltoall
CG	conj_grad()	MPI_Send	MPI_Wait
MG	resid()	mg3P()	MPI_Send

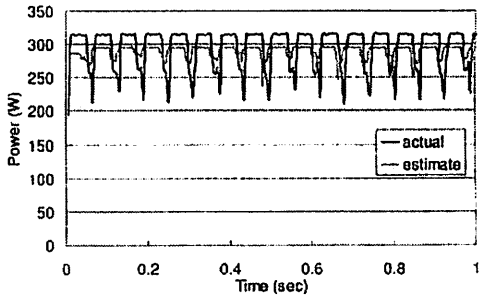

図 6 Turion cluster における消費電力の推測と実測の比較 (NPB FT 1.8GHz)

4.3 ベンチマークプログラムと領域

評価には，NAS Parallel Benchmarks のバージョン 3.2.1 の中から FT，IS，CG，MG を使用する．問題サイズは C を使用し，MPI ライブラリには LAM 7.1.1 を使用した．アプリケーションの領域の分割は，我々の先行研究と同様に main 関数から呼び出される主要な関数および，MPI 通信を領域とした．表 3 に各ベンチマークにおける主な領域を示す．

4.4 消費電力の推測

各ベンチマークを，実行可能な全ての周波数で実行し，全ノードの実行プロファイルと全体の消費電力プロファイルを取得し，消費電力の推測を行った．図 2 と図 3 に周波数を 2.0GHz としたときの，FT と IS を実行したときのシステム全体の実測と推測した各ノードの消費電力の合計の変遷を示す．どちらのベンチマークにおいても，消費電力の特性をうまくとらえることができている，

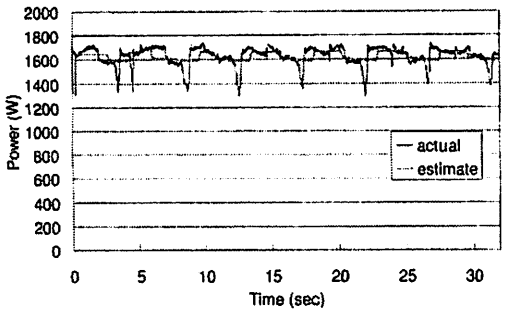

図 2 NPB FT における実測と推測消費電力 (2.0GHz)

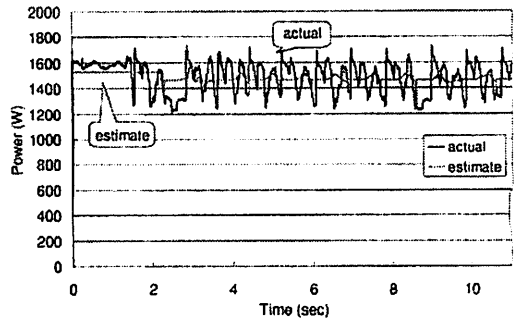

図 3 NPB IS における実測と推測消費電力 (2.0GHz)

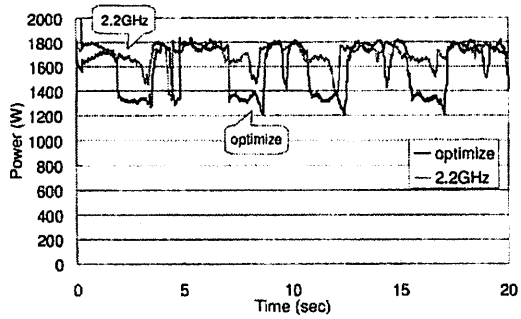

図 4 NPB FT における実測と推測消費電力

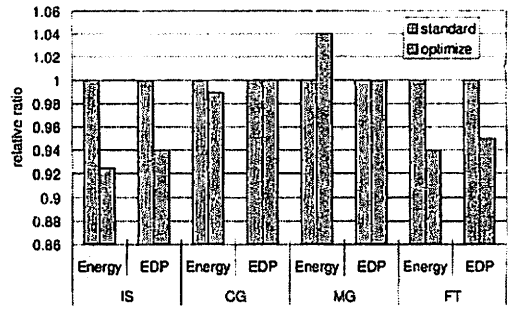

図 5 最適化した時の標準動作周波数と比較した PDP，EDP 比

4.5 周波数選択アルゴリズムの適用

推測から求められた各ノードの消費電力を使用し，各領域を最適に実行する周波数を選択する．周波数の選択には，我々の先行研究で提案したアルゴリズムを使用する．このアルゴリズムは，DVFS を使用するときに問題となる周波数変更に必要なオーバーヘッドを考慮しており，単純にエネルギーの大小比較を行っているものではない．また，様々な評価指標を用いてスケジューリングを行うことができる．ここでは，エネルギーである PDP と電力遅延積である EDP の 2 つの指標を最適にするスケジューリングを行った．表 4 にアルゴリズムが選択した各領域の周波数を示す．この結果から以下のことがわかる

- 全体全通信の場合，低い周波数を選択する
- 計算を行う領域では，概ね高い周波数を選択する
- EDP の場合，CG と MG では最も高い周波数に固定となる

この周波数を各領域に適用し，実測を行った．図?? に FT と IS における最高周波数に固定した場合と周波数スケジューリングを適用した場合の消費電力の比較を示す．このように，消費電力を下げることができることがわかる．図 5 に周波数スケジューリングを行った場合の各ベンチマークの PDP と EDP の削減の効果を示す．PDP に関しては，IS で約 9%，FT で約 5%のエネルギーの削減を行うことができた．一方で，CG ではほぼ同じになり，MG ではエネルギー消費が増大した．この原因を特定するために，MG において，詳細に各領域を調べると，*MPI_send* の領域の実行時間が，急上昇していることがわかった．この領域は，細かい通信をくり返し行っており，今回のスケジューリングにおいて選択した DVFS のオーバーヘッド，50 μ 秒が十分な値ではなかった可能性がある．

EDP に関しては，IS では約 8%，FT では約 4%の削減を実現した．一方で，CG や MG などでは，常に最高周波数を選択する結果となり，削減を行うことができなかった．この理由として以下のことが考えられる

- EDP はある区間のエネルギーとその区間の実行時間の積となるため，今回のような同じ領域であれば，消費電力が常に同一であるとした場合，求められる EDP が異なる可能性がある．
- CG や MG などでは，1 対 1 通信が非常に多く，また，1 回あたりの通信時間も短いために周波数変更によるオーバーヘッドが実行時間に大きな影響を与え EDP を悪化させる．

表 4 アルゴリズムが決定した周波数 [MHz]

	region	rank	allreduce	alltoallv
IS	PDP	2000	2200	1000
	EDP	2000	2200	1400
	region	fft	evolve	alltoall
FT	PDP	2200	2200	1000
	EDP	2200	2200	1000
	region	conj-grad	MPIsend	MPIwait
CG	PDP	2000	2000	2000
	EDP	2200	2200	2200
	region	resid	mg3P	MPIsend
MG	PDP	2200	2000	1800
	EDP	2200	2200	2200

5. 考　　察

5.1 電力性能最適化について

各ノードの消費電力の測定が困難である大規模クラスタシステムにおいて，最適に DVFS を使用するため，全体消費電力プロファイルを使用した最適化を提案した．システム全体の消費電力と各ノードの実行プロファイルから，各ノードの消費電力プロファイルを推測することで，各ノードの周波数を最適にスケジューリングすることができた．周波数スケジューリングを適用した結果，PDP で最大約 9%，EDP で最大約 8%の削減を行うことができた．今回評価を行ったシステムは，プロセッサ以外の消費電力が大きいため，idle 時と計算時の消費電力の差がそれほど大きくなかったが，大きなエネルギーの削減を実現できた．今回はベンチマークに NPB を使用したため，各ノードの負荷のバランスがとれており，また概ね全てのノードは同時に同じ領域を実行するため，消費電力の推測を高い精度で行うことができた．

5.2 スケーラビリティについて

数千ノードを越えるシステムでは，システム全体の消費電力を測定するのも困難であるため，我々は現在，小規模のクラスタシステムにおける結果を元に，大規模システムの消費電力を下げる手法の検討を行っている．V.Freeh[7] らはクラスタシステムにおける DVFS スケジューリングのスケーラビリティに関する研究として，周波数と通信のノード数に応じた予測モデルを作ることで消費電力の削減を実現している．我々は，今後の課題として本論文で提案した全体消費電力プロファイルからの各ノードの消費電力プロファイル予測手法と先行研究である周波数選択アルゴリズムに，このような予測モデルを適用することでより高い消費電力の削減を実現できる可能性がある．

6. おわりに

本論文では，大規模な HPC クラスタシステムにおいて DVFS による消費電力の削減を実現するために，システム全体の消費電力プロファイルを元にした周波数選択の最適化の検討を行った．最適化において，いくつかの領域に分けたアプリケーションを可能な全ての周波数で実行することで，システム全体の消費電力と各ノードの実行プロファイルを取得しした．得られたプロファイル情報を元に各領域の実行時間とエネルギーの関係式を作成し，各領域の平均消費電力を求めることで各ノードの消費電力プロファイルを推測した．推測した消費電力プロファイルを使用し，過去に我々が提案した周波数選択アルゴリズムを適用することで，それぞれの領域を最適に実行する動作周波数を求めた．この結果，システム全体で PDP は最大約 9%，EDP は最大約 8%の削減を実現した．

今後の課題として，様々なアプリケーションにおいて各ノードの消費電力の予測を行う．また，アルゴリズムのスケーラビリティについても検討を行い，より大規模なシステムの消費電力の削減を実現することで管理コストや故障に対するコストなどの削減をすることが期待できる．

謝　　辞

様々な御助言をいただいた CREST チームの方々に感謝します．本研究の一部は，科学技術振興機構・戦略的創造研究「低消費電力化とモデリング技術によるメガスケールコンピューティング」による．

参 考 文 献

1) N. Adiga, G. Almasi, G. Almasi et.al. An Overview of the BlueGene/L Supercomputer. In *Supercomputing Conference 05 (SC'05)*, November 2005.

2) AMD. Corp. AMD Opteron^{TM} Processor Power and Thermal Data Sheet, 2006.

3) Transmeta Corp. Longrun Thermal Management, 2001. htto://www.transmeta.com/

4) H.Nakashima, H.Nakamura, M.Sato, T.Boku, S.Matsuoka, D.Takahashi, and Y.Hotta. Megaproto: 1 TFlops/10kw rack is feasible even with only commodity technology. In *Supercomputing Conference 05*, November 2005.

5) M.Warren, E.Weigle, and W.Feng. High-Density Computing: A 240-processor Beowulf in One Cubic Meter. In *Supercomputing Conference 02*, November 2002.

6) Y.Hotta, M.Sato, H.Kimura et.al. Profile-based Optimization of Power Performance on by using Dynamic Voltage Scaling on a PC cluster In *2nd High Performance Power-Aware Computing Workshop(HPPAC) in IPDPS'06*, April 2006.

7) R.Springer and D.K.Lowenthal et.al. Minimizing execution time in MPI programs on an energy-constrained, power-scalable cluster In *PPoPP'06: Proceedings of the 11th ACM SIGPLAN symposium on Principles and practice of parallel programming* , March 2006.